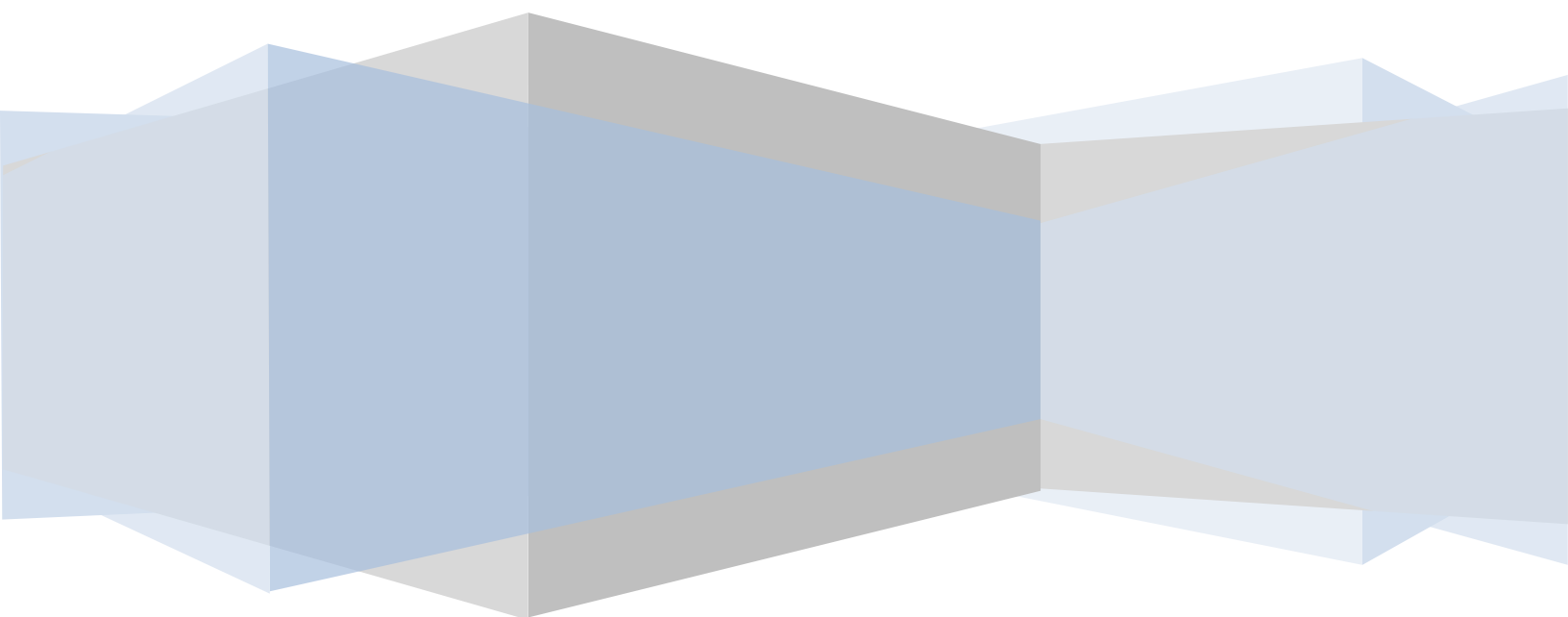


# Xen Virtualization Essentials

Virtualization Deployment and Management



Xen Virtualization Essentials – First Edition

© 2009 Virtuatopia.com. This eBook is provided for personal use only. Unauthorized use, reproduction and/or distribution strictly prohibited. All rights reserved.

## Table of Contents

Chapter 1. About Xen Virtualization Essentials .....	10
1.1 What is Virtualization? .....	10
1.2 Why is Virtualization Important? .....	10
Chapter 2. An Overview of Virtualization Techniques .....	13
2.1 Guest Operating System Virtualization .....	13
2.2 Shared Kernel Virtualization .....	14
2.3 Kernel Level Virtualization .....	15
2.4 Hypervisor Virtualization.....	16
2.5 Paravirtualization .....	17
2.5.1 Full Virtualization .....	17
2.5.2 Hardware Virtualization.....	17
Chapter 3. Configuring and Installing a Xen Hardware Virtual Machine (HVM) Guest.....	19
3.1 Checking Hardware Support for Xen Hardware Virtual Machines (HVM).....	19
3.2 Preparing to Install a Xen HVM domainU Guest.....	19
3.3 Creating a Xen HVM Configuration File .....	20
3.4 Booting the HVM Guest .....	26
3.5 Connecting to the HVM domainU Guest Graphical Console .....	26
Chapter 4. Installing and Running Windows XP or Vista as a Xen HVM domainU Guest .....	28
4.1 Pre-Requisites for Installing a Windows Xen Guest.....	28
4.2 Preparing to Install Windows.....	28
4.3 Preparing the Windows XP / Windows Vista for Xen HVM Installation .....	29
4.4 Configuring the Xen Windows Guest Configuration File .....	29
4.5 Starting the Xen Windows HVM Guest .....	33
Chapter 5. Virtualizing Windows Server 2008 with Xen.....	34
5.1 Requirements for Xen Windows Server 2008 Virtualization .....	34
5.2 Windows Server 2008 Installation Media .....	34
5.3 Preparing Storage Space for Windows Server 2008 .....	35

5.4	Creating the Xen Windows Server 2008 Configuration File.....	35
5.5	Starting the Xen Windows Server 2008 HVM Guest.....	38
Chapter 6.	Installing and Running Windows 7 as a Xen HVM domainU Guest .....	40
6.1	Requirements for Xen Windows 7 Virtualization.....	40
6.2	Windows 7 Installation Media .....	40
6.3	Preparing Storage Space for Windows 7.....	41
6.4	Creating the Xen Windows 7 Configuration File .....	41
6.5	Starting the Xen Windows 7 HVM Guest .....	44
Chapter 7.	Adding USB Devices to a Xen HVM domainU Guest.....	45
7.1	Identifying USB Devices on the Xen Host System .....	45
7.2	A Trick to Identify the Correct USB Device.....	45
7.3	Adding the New USB Device to the Xen domainU Configuration File .....	46
7.4	Temporarily Adding a USB Device to a running Xen HVM Guest .....	46
Chapter 8.	Building a Xen Virtual Guest File System on a Disk Image (Cloning Host System) .	48
8.1	Xen Requirements.....	48
8.2	Creating a Disk Image for the Root File system .....	48
8.3	Creating a Swap Space Disk Image.....	49
8.4	Cloning the Host OS on the Guest Domain .....	50
8.5	Creating a Xen Configuration File.....	51
8.6	Configuring System Files for the Guest Operating System .....	53
8.7	Modifying /etc/fstab for the Guest System .....	53
8.8	Booting the Guest OS.....	54
Chapter 9.	Building a Xen Virtual Guest File System on a Physical Disk Partition (Cloning Host System) .....	56
9.1	An Overview of the Xen Host and Guest Physical Disks .....	56
9.2	Preparing the Xen Disk Partitions .....	56
9.3	Creating the a File System on the Xen Guest Root Partition .....	59
9.4	Configuring the Swap Partition for the Xen Guest System .....	59

9.5	Mounting the Root File System.....	60
9.6	Cloning the Host OS on the Guest Root Partition .....	60
9.7	Creating a Xen Configuration File.....	61
9.8	Configuring System Files for the Guest Operating System .....	63
9.9	Modifying /etc/fstab for the Guest System .....	64
9.10	Booting the Guest OS .....	65
Chapter 10.	Building a Xen Virtual Guest File System using Logical Volume Management (LVM) .....	67
10.1	The Key Components of Logical Volume Management.....	67
10.1.1	Volume Group (VG).....	67
10.1.2	Physical Volume (PV) .....	67
10.1.3	Logical Volume (LV).....	67
10.1.4	Physical Extent (PE).....	67
10.1.5	Logical Extent (LE).....	67
10.2	Preparing for an LVM based Xen Guest Domain.....	68
10.3	Converting Physical Disks into Physical Volumes.....	68
10.4	Creating a Volume Group.....	69
10.5	Creating a Logical Volume for the Xen Guest System.....	70
10.6	Creating a File System on the Logical Volume .....	71
10.7	Configuring the Swap Partition for the Xen Guest System .....	72
10.8	Mounting the Root File System.....	72
10.9	Cloning the Host OS on the Guest Root Partition .....	73
10.10	Creating a Xen Configuration File.....	74
10.11	Configuring System Files for the Guest Operating System .....	75
10.12	Modifying /etc/fstab for the Guest System.....	76
10.13	Booting the Guest OS .....	77
Chapter 11.	Building a Xen Guest Root File System using yum and rpm .....	79
11.1	An Overview of Repository Based Installation.....	79

11.2	Preparing to Create a Xen Guest Root File System from Repositories .....	79
11.3	Beginning the Installation .....	80
11.4	Configuring the Guest System using chroot.....	81
11.5	Booting the Xen Guest System.....	81
Chapter 12. Building a Debian or Ubuntu Xen Guest Root File System using debootstrap.....		83
12.1	Creating the Xen Guest Root File System .....	83
12.2	Creating Swap for the Xen Guest Domain .....	84
12.3	Installing the Base Ubuntu/Debian System using debootstrap.....	85
12.4	Configuring the root Password .....	86
12.5	Creating a Configuration File for the Guest Domain.....	86
12.6	Configuring System Files for the Guest Operating System .....	88
12.7	Modifying /etc/fstab for the Guest System .....	89
12.8	Booting the Guest OS.....	89
Chapter 13. Building a Xen Guest Domain using Xen-Tools.....		90
13.1	Getting Xen-Tools.....	90
13.2	Configuring Xen-Tools .....	90
13.3	Specify Xen-Tools Installation Location .....	90
13.4	Specifying the Xen-Tools Installation Source and Method .....	91
13.5	Configuring Disk Space and Memory for the Xen Guest.....	91
13.6	Choosing a Linux Distribution .....	91
13.7	Configuring Xen Guest Network Options.....	92
13.8	Configuring the Kernel and RAM Disk.....	92
13.9	Defining the Installation Source.....	93
13.10	Miscellaneous Settings Xen-Tools Configuration .....	93
13.11	Xen Guest Console Settings.....	93
13.12	Options Disk Drive Device Naming.....	94
13.13	Building the Xen Guest Images.....	94
13.14	Booting the Xen Guest System .....	96

Chapter 14. Using QEMU Disk Images for Xen DomainU Systems .....	97
14.1 Creating a QEMU Disk Image for the Xen domainU .....	97
14.2 Installing the Guest Operating System.....	97
14.3 Configuring the Guest Operating System for Xen.....	97
14.4 Mounting QEMU Disk Image Partitions .....	99
14.5 pyGRUB and the Xen Configuration File .....	99
14.6 Booting the Xen domainU System .....	100
Chapter 15. Creating and Booting a Xen Guest domainU using an NFS Mounted Root File system .....	101
15.1 Kernel Requirements for NFS based Root File systems .....	101
15.2 Populating the Root File system .....	101
15.3 Exporting and Mounting the Xen Guest Root File system .....	102
15.4 Creating the Xen Configuration File for the NFS Root File system .....	102
15.5 Booting the Xen domainU .....	103
Chapter 16. Configuring a VNC based Graphical Console for a Xen Paravirtualized domainU Guest .....	105
16.1 What is VNC?.....	105
16.2 VNC Security.....	105
16.3 Configuring a Xen domainU for VNC Access .....	106
16.4 Connecting to the Guest Desktop using VNC Viewer .....	106
16.5 Starting a Graphical Desktop on the Xen domainU Guest.....	108
16.6 Establishing a Secure Remote Desktop Session.....	109
Chapter 17. Running and Connecting to VNC Servers on a Xen Guest (domainU) System .....	111
17.1 Installing VNC on the Xen domainU Guest .....	111
17.2 Creating Xen domainU VNC Desktop Sessions.....	111
17.3 Connecting to a Xen domainU Remote Desktop .....	112
17.4 Configuring the Xen domainU Desktop Environment.....	113
17.5 Closing Down a domainU VNC Desktop Session .....	116

Chapter 18. Adding Disk, CDROM and DVD Devices to a Running Xen domainU Guest System ...	118
18.1 Requirements for Xen domainU Block Device Attachment .....	118
18.2 An Overview of <i>xm block-attach</i> .....	118
18.3 Attaching a Device to a domainU Guest .....	119
18.4 Mounting the Device in the domainU Guest .....	120
18.5 Detaching a Device from the domainU Guest .....	120
Chapter 19. Xen Monitoring Tools and Techniques .....	121
19.1 Why Monitor a Xen Environment? .....	121
19.2 Obtaining Xen Configuration and System Information .....	121
19.3 Monitoring Xen Performance with XenMon .....	122
19.4 Monitoring Performance with XenTop .....	124
Chapter 20. Migrating Xen domainU Guests Between Host Systems .....	126
20.1 Requirements for Xen domainU Migration .....	126
20.2 Enabling Xen Guest Migration .....	126
20.3 Xen Migration Firewall Configuration .....	127
20.4 Preparing the Xen Migration Environment .....	127
20.5 Running the DomainU Guest .....	128
20.6 Performing the Migration .....	129
20.7 Checking the Xen Log for Migration Errors .....	130
Chapter 21. Solving Common Xen Problems .....	133
21.1 Xen guest domainU migration fails with Error: can't connect: No route to host message .....	133
21.2 Windows Server 2008 Xen HVM installation fails - Firmware (BIOS) is not ACPI compatible .....	134
21.3 Xen mouse pointer appears in the wrong position in VNC console .....	136
21.3.1 Problem Cause .....	136
21.3.2 Changing the Xen domainU Configuration .....	136
21.3.3 Resolving the Problem using Windows Settings .....	136



21.4 Device not visible in Xen domainU guest when using xm block-attach command to add a disk or CD-ROM/DVD drive ..... 137

21.5 Xen domainU Guest has an IP address on 192.168.122 subnet instead of the subnet to which the domain0 host belongs ..... 138

21.6 Ubuntu domainU Fails to boot with "Error: Device 0 (vif) could not be connected. Could not find bridge, and none was specified" ..... 141

21.7 Ubuntu Xen Guest (DomU) Hangs after EXT3-fs: mounted file system with ordered data mode message ..... 142

21.8 Ubuntu Xen System Boot Hangs After Setting System Clock Message ..... 144

21.9 Xen domainU fails to boot with a "Xen Guest OS Fails to Boot with Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(0,0)" error message ..... 145

21.10 A Xen Guest OS fails to boot with a "switchroot: mount failed: No such file or directory error message" error message ..... 146

21.11 Xen CentOS/Fedora/Red Hat Guest OS Hangs during Boot ..... 147

21.12 Xen domainU Boot Fails with Invalid kernel / ERROR: Not a Xen-ELF image Message .  
..... 151

## Chapter 1. About Xen Virtualization Essentials

---

Though the concept of virtualization is far from new, recent requirements such as the need to maximize hardware utilization, decrease hardware costs, reduce power consumption and simplify system management and security have led to a significant increase in both the deployment of virtualization and the number of available virtualization solutions. In fact, virtualization solutions are now available to meet just about every need, from the global enterprise all the way down to the home user.

This book is about one such virtualization solution known as Xen. Xen is a feature rich, open source, hypervisor-based virtualization solution which, in spite of its relatively recent origins, has gained both wide acceptance and an enviable reputation throughout the IT industry.

The objective of this book is to provide the reader with an understanding of the basic approaches to virtualization together with detailed information on deploying virtualization using Xen technology. Whilst many books tend to focus on the theory of virtualization, this book emphasizes the practical aspects of working with Xen, including detailed step by step tutorials designed to show exactly how to create, deploy and manage Xen based guest domains.

### 1.1 What is Virtualization?

In a traditional computing model, a computer system typically runs a single operating system. For example, a desktop computer might run a copy of Windows XP or Windows Vista, while a server might run Linux or Windows Server 2008. The concept of virtualization, as it pertains to this book, involves the use of a variety of different technologies to allow multiple and potentially varied operating system instances to run concurrently on a single physical computer system, each sharing the physical resources of the host computer system (such as memory, network connectivity, CPU and storage). Within a virtualized infrastructure, a single physical computer server might, for example, run two instances of Windows Server 2008 and one instance of Linux. This, in effect, allows a single computer to provide an IT infrastructure that would ordinarily require multiple computer systems.

### 1.2 Why is Virtualization Important?

Virtualization has gained a considerable amount of coverage in the trade media in recent years. Given this sudden surge of attention it would be easy to make the assumption that the concept of virtualization is new. In fact, virtualization has been around in one form or another since it was first introduced on IBM mainframe operating systems in the 1960s. The reason for the

sudden popularity of virtualization can be attributed to a number of largely unconnected trends:

- **Green computing** - So called *green computing* refers to the recent trend to reduce the power consumption of computer systems. Whilst not a primary concern for individual users or small businesses, companies with significant server operations can save considerable power usage levels by reducing the number of physical servers required using virtualization. An additional advantage involves the reduction in power used for cooling purposes, since fewer servers generate less heat.
- **Increased computing power** - The overall power of computer systems has increased exponentially in recent decades to the extent that many computers, by running a single operating system instance, are using a fraction of the available memory and CPU power. Virtualization allows companies to maximize utilization of hardware by running multiple operating systems concurrently on single physical systems.
- **Financial constraints** - Large enterprises are under increasing pressure to reduce overheads and maximize shareholder returns. A key technique for reducing IT overheads is to use virtualization to gain maximum return on investments in computer hardware.
- **Web 2.0 & Cloud Computing** - The term Web 2.0 has primarily come to represent the gradual shift away from hosting applications and data on local computer systems to a web based approach. For example, many users and companies now use Google Apps for spreadsheet and word processing instead of installing office suite software on local desktop computers. Web services such as these require the creation of vast server farms running hundreds or even thousands of servers, consuming vast amounts of power and generating significant amounts of heat. Virtualization allows web services providers to consolidate physical server hardware, thereby cutting costs and reducing power usage.
- **Operating system market fragmentation** - In recent years the operating system market has increasingly fragmented with Microsoft ceding territory to offerings such as Linux and Apple's Mac OS. Enterprises are now finding themselves managing heterogeneous environments where, for example, Linux is used for hosting web sites whilst Windows Server is used to email and file serving functions. In such environments, virtualization allows different operating systems to run side by side on the same computer systems. A similar trend is developing on the desktop, with many users considering Linux as an alternative Microsoft Windows. Desktop based virtualization allows users to run both

Linux and Windows in parallel, a key requirement given that many users looking at Linux still need access to applications that are currently only available on Windows.

## Chapter 2. An Overview of Virtualization Techniques

---

Throughout this book the word *virtualization* is used within the context of using Xen technology to run multiple operating systems on a single physical computer system. It is important, however, to appreciate that virtualization is actually a "catch all" term that refers to a variety of different solutions and technologies, of which Xen is only one.

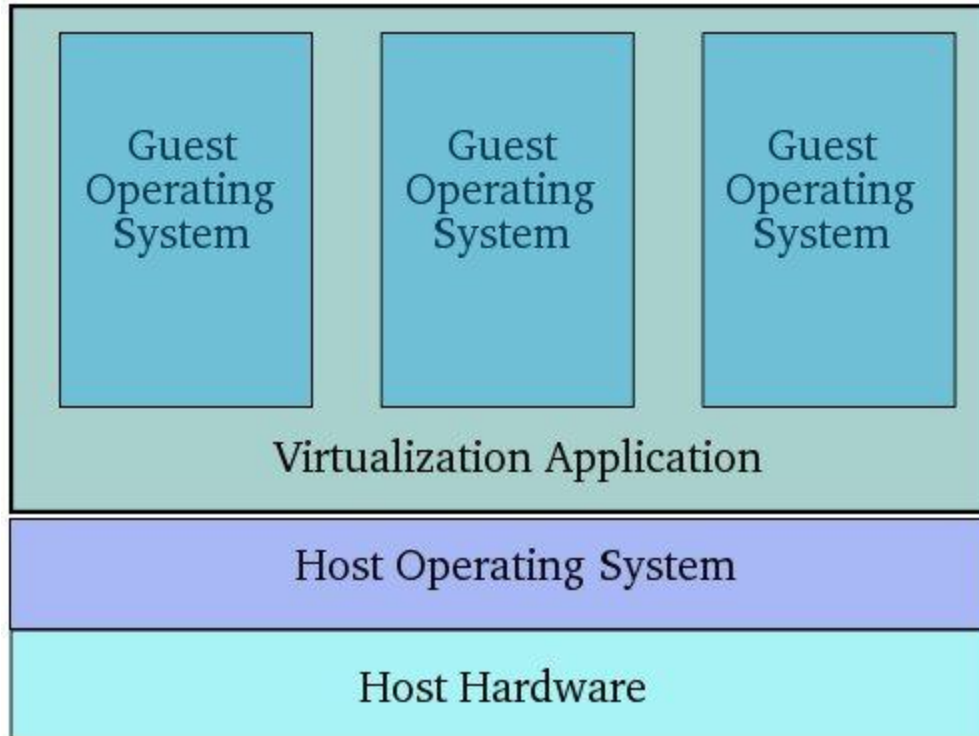
When deciding on the best approach to implementing virtualization it is important to have a clear understanding of the different virtualization solutions that are currently available. The purpose of this chapter, therefore, is to describe in general terms the four virtualization techniques in common use today, namely guest operating system, shared kernel, hypervisor and kernel level.

### 2.1 Guest Operating System Virtualization

Guest OS virtualization, also referred to as application based virtualization, is perhaps the easiest concept to understand. In this scenario the physical host computer system runs a standard unmodified operating system such as Windows, Linux, UNIX or Mac OS X. Running on this operating system is a virtualization application which executes in much the same way as any other application such as a word processor or spreadsheet would run on the system. It is within this virtualization application that one or more virtual machines are created to run the guest operating systems on the host computer. The virtualization application is responsible for starting, stopping and managing each virtual machine and essentially controlling access to physical hardware resources on behalf of the individual virtual machines. The virtualization application also engages in a process known as *binary rewriting* which involves scanning the instruction stream of the executing guest system and replacing any privileged instructions with safe emulations. This has the effect of making the guest system think it is running directly on the system hardware, rather than in a virtual machine within an application.

An example of guest OS virtualization technology is VMware Server. VirtualBox also uses application virtualization although it also provides support for hardware virtualization.

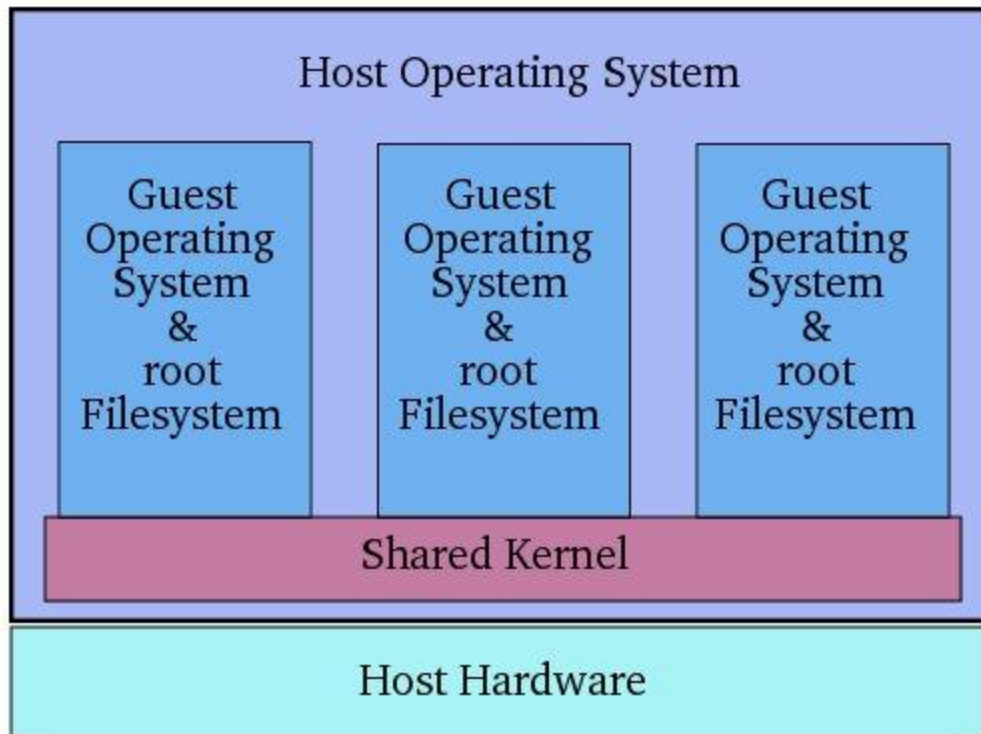
The following figure provides an illustration of guest OS based virtualization:



As outlined in the above diagram, the guest operating systems operate in virtual machines within the virtualization application which, in turn, runs on top of the host operating system in the same way as any other application. Clearly, the multiple layers of abstraction between the guest operating systems and the underlying host hardware are not conducive to high levels of virtual machine performance. This technique does, however, have the advantage that no changes are necessary to either host or guest operating systems and no special CPU hardware virtualization support is required.

## 2.2 Shared Kernel Virtualization

Shared kernel virtualization (also known as system level or operating system virtualization) takes advantage of the architectural design of Linux and UNIX based operating systems. In order to understand how shared kernel virtualization works it helps to first understand the two main components of Linux or UNIX operating systems. At the core of the operating system is the *kernel*. The kernel, in simple terms, handles all the interactions between the operating system and the physical hardware. The second key component is the *root file system* which contains all the libraries, files and utilities necessary for the operating system to function. Under shared kernel virtualization the virtual guest systems each have their own *root file system* but share the kernel of the host operating system. This structure is illustrated in the following architectural diagram:



This type of virtualization is made possible by the ability of the kernel to dynamically change the current root file system (a concept known as *chroot*) to a different root file system without having to reboot the entire system. Essentially, shared kernel virtualization is an extension of this capability. Perhaps the biggest single drawback of this form of virtualization is the fact that the guest operating systems must be compatible with the version of the kernel which is being shared. It is not, for example, possible to run Microsoft Windows as a guest on a Linux system using the shared kernel approach. Nor is it possible for a Linux guest system designed for the 2.6 version of the kernel to share a 2.4 version kernel.

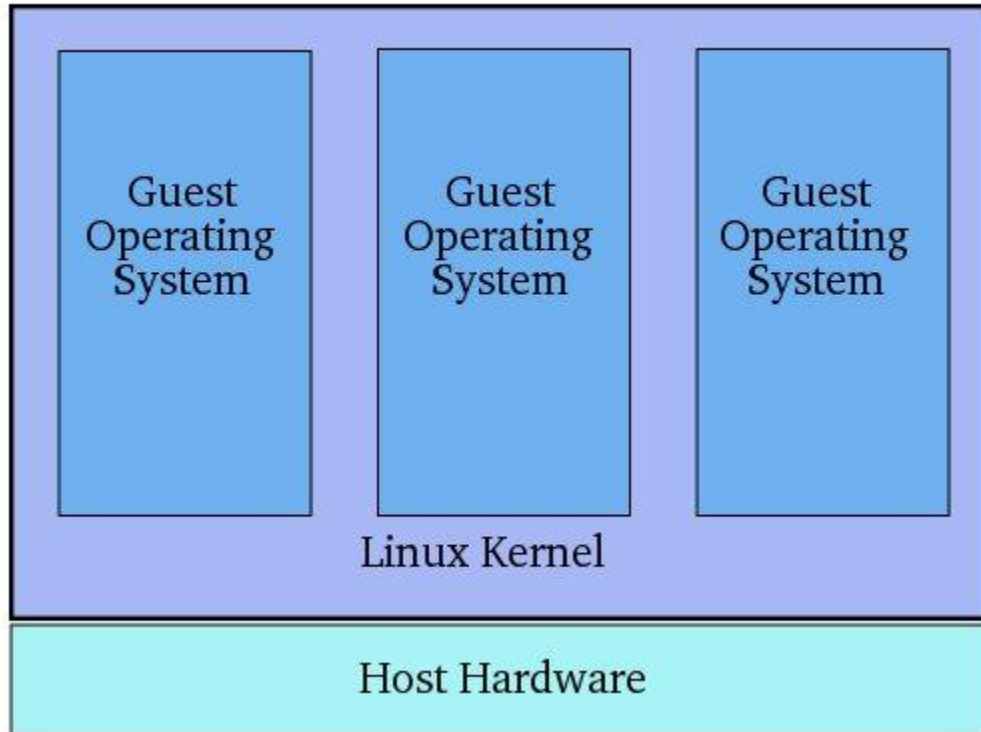
Linux VServer, Solaris Zones and Containers, FreeVPS and OpenVZ are all examples shared kernel virtualization solutions.

### 2.3 Kernel Level Virtualization

Under kernel level virtualization the host operating system runs on a specially modified kernel which contains extensions designed to manage and control multiple virtual machines each containing a guest operating system. Unlike shared kernel virtualization each guest runs its own kernel, although similar restrictions apply in that the guest operating systems must have been compiled for the same hardware as the kernel in which they are running. Examples of kernel

level virtualization technologies include User Mode Linux (UML) and Kernel-based Virtual Machine (KVM).

The following diagram provides an overview of the kernel level virtualization architecture:



## 2.4 Hypervisor Virtualization

The x86 family of CPUs provide a range of *protection levels* also known as *rings* in which code can execute. Ring 0 has the highest level privilege and it is in this ring that the operating system kernel normally runs. Code executing in ring 0 is said to be running in *system space*, *kernel mode* or *supervisor mode*. All other code such as applications running on the operating system operates in less privileged rings, typically ring 3.

Under hypervisor virtualization a program known as a *hypervisor* (also known as a type 1 Virtual Machine Monitor or VMM) runs directly on the hardware of the host system in ring 0. The task of this hypervisor is to handle resource and memory allocation for the virtual machines in addition to providing interfaces for higher level administration and monitoring tools.

Clearly, with the hypervisor occupying ring 0 of the CPU, the kernels for any guest operating systems running on the system must run in less privileged CPU rings. Unfortunately, most operating system kernels are written explicitly to run in ring 0 for the simple reason that they need to perform tasks that are only available in that ring, such as the ability to execute



privileged CPU instructions and directly manipulate memory. A number of different solutions to this problem have been devised in recent years, each of which is described below:

## 2.5 Paravirtualization

Under paravirtualization the kernel of the guest operating system is modified specifically to run on the hypervisor. This typically involves replacing any privileged operations that will only run in ring 0 of the CPU with calls to the hypervisor (known as *hypercalls*). The hypervisor in turn performs the task on behalf of the guest kernel. This typically limits support to open source operating systems such as Linux which may be freely altered and proprietary operating systems where the owners have agreed to make the necessary code modifications to target a specific hypervisor. These issues notwithstanding, the ability of the guest kernel to communicate directly with the hypervisor results in greater performance levels compared to other virtualization approaches.

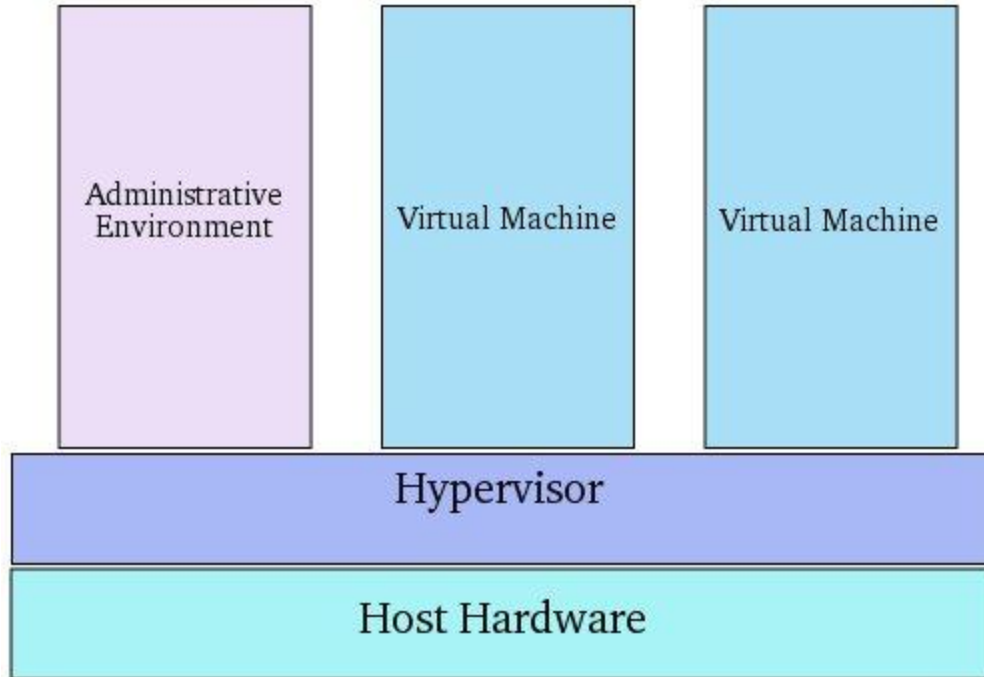
### 2.5.1 Full Virtualization

Full virtualization provides support for unmodified guest operating systems. The term *unmodified* refers to operating system kernels which have not been altered to run on a hypervisor and therefore still execute privileged operations as though running in ring 0 of the CPU. In this scenario, the hypervisor provides CPU emulation to handle and modify privileged and protected CPU operations made by unmodified guest operating system kernels. Unfortunately this emulation process requires both time and system resources to operate resulting in inferior performance levels when compared to those provided by paravirtualization.

### 2.5.2 Hardware Virtualization

Hardware virtualization leverages virtualization features built into the latest generations of CPUs from both Intel and AMD. These technologies, known as Intel VT and AMD-V respectively, provide extensions necessary to run unmodified guest virtual machines without the overheads inherent in full virtualization CPU emulation. In very simplistic terms these new processors provide an additional privilege mode above ring 0 in which the hypervisor can operate essentially leaving ring 0 available for unmodified guest operating systems.

The following figure illustrates the hypervisor approach to virtualization:



As outlined in the above illustration, in addition to the virtual machines, an administrative operating system and/or management console also runs on top of the hypervisor allowing the virtual machines to be managed by a system administrator. Hypervisor based virtualization solutions include Xen, VMware ESX Server and Microsoft's Hyper-V technology.